# On genetic algorithms minimizing a class of FSA with fuzzy automata

A.V. Kelarev[1]
*School of Computing, University of Tasmania*
*Private Bag 100, Tasmania 7001, Australia*
*E-mail: Andrei.Kelarev@utas.edu.au*

**Abstract.** Finite state automata are crucial for numerous practical algorithms of computer science. We show how to use genetic algorithms and fuzzy automata to simplify a class of FSA defined by labeled graphs and considered in the literature.

Finite state automata, FSA, are well-known tools used in coding theory, text processing, image analysis and compression, speech recognition, and bioinformatics. Labeled directed graphs have been applied in [2], [4] and [5] to define and investigate a class of FSA.

Throughout the word *graph* means a finite directed graph without multiple edges but possibly with loops, and $D = (V, E)$ is a graph. Graphs have been used by several authors to define automata and investigate properties of languages accepted by them, see [2]. A *language* over an alphabet $X$ is a subset of the free monoid $X^*$ generated by $X$. For standard concepts of automata and languages theory we refer to [2] and [8].

Let $X$ be an alphabet, $f : X \to V$ a mapping, and let $T$ be a subset of $V \cup \{0, 1\}$. The FSA $\mathrm{Atm}_\ell(D) = \mathrm{Atm}_\ell(D, T) = \mathrm{Atm}_\ell(D, T, f)$ of the graph $D$ is the finite state acceptor with

(LA1) the set of states $V \cup \{0, 1\}$;

(LA2) the initial state 1;

(LA3) the set of terminal states $T$;

(LA4) the next-state function given, for a state $u$ and a letter $x \in X$, by defining $u \cdot x$ to be equal to $f(x)$, if $(f(x), u) \in E$ or $u = 1$, and be 0 otherwise.

For motivation and relations to previous results the reader is referred to [2]. A *language* over $X$ is a set of words that can be formed by the letters of $X$, i.e., a subset of the free monoid $X^*$ generated by $X$. The *language recognized* or *accepted* by $\mathrm{Atm}_\ell(D, T)$ is the set $\{u \in X^* \mid 1 \cdot u \in T\}$.

Combinatorial minimization algorithms for FSA of this type are computationally expensive. Efficient optimization methods, like those based on genetic algorithms, are not directly applicable either, since randomized steps of these algorithms can dramatically change the crisp language recognized by the FSA. In order to apply genetic algorithms for minimizing the FSA, we offer the following general scheme, where it is suggested to replace the original crisp FST with its fuzzy analogue, as summarized in Figure 1. For preliminaries on fuzzy systems theory and genetic algorithms we refer to [1], [6], and [7].

**Algorithm 1** *Enables the application of genetic algorithms to the minimization of FSA* $\mathrm{Atm}_\ell(D, T, f)$ *with fuzzy automata.*

Step 1. Replace $\mathrm{Atm}_\ell(D, T, f)$ with equivalent $\mathrm{fAtm}_\ell(D, T, f)$.
Step 2. Encode each fuzzy set of $\mathrm{fAtm}_\ell(D, T, f)$ a minimal set of parameters.
Step 3. Define crossover operation on blocks of the encoding.
Step 4. Use a genetic algorithm to minimize $\mathrm{fAtm}_\ell(D, T, f)$.
Step 5. Defuzzify the minimal fuzzy automaton.

Figure 1: The general scheme of applying genetic algorithms via fuzzy automata.

Fuzzy automata have been investigated by many authors (see [6] for references). This

[1]Supported by Discovery grant DP0449469 from the Australian Research Council.

concept is related to the notion of a fuzzy language, that has been introduced explicitly by Zadeh [10]. Next, we follow [3] and [6] and define the concept of fuzzy automaton that can be used to approximate the functions of crisp automata above. The required standard concepts of fuzzy theory, like fuzzy congruences, etc., are explained in [6].

A fuzzy automaton is a system $\mathbf{A} = (S, \Lambda, \overline{p}, F, G)$, where $S$ is a finite set of states; $\Lambda$ is a finite set of inputs; $\overline{p} \underset{\sim}{\subseteq} S$ is a fuzzy set called a *fuzzy initial state*; $F : S \times \Lambda \times S \to [0,1]$ is a fuzzy transition function, i.e., $F(\lambda) \underset{\sim}{\subseteq} S \times S$ is a fuzzy transition matrix; and $G \underset{\sim}{\subseteq} S$ is a fuzzy set called a *fuzzy final state*.

For $s, s' \in S$ and $\lambda \in \Lambda$, recall that $F(s, \lambda, s')$ denotes the grade of transition of a fuzzy automaton from state $s$ to state $s'$ when the input is $\lambda \in \Lambda$.

If we denote the free monoid generated by $\Lambda$ by $\Lambda^*$, then $F$ can be extended to a fuzzy transition function

$$F^* : S \times \Lambda^* \times S \to [0,1],$$

where $\lambda = \lambda_1 \lambda_2 \ldots \lambda_n \in \Lambda^*$. Then the following diagram commutes

$$\begin{array}{ccc} S \times \Lambda \times S & \xrightarrow{F} & [0,1] \\ \mu \big\downarrow & & \big\| \\ S \times \Lambda^* \times S & \xrightarrow{F^*} & [0,1] \end{array}$$

Here, $\mu$ is an embedding of $S \times \Lambda \times S$ into $S \times \Lambda^* \times S$, given by

$$F^*(\lambda_1 \lambda_2 \ldots \lambda_n) = F(\lambda_1) \circ F(\lambda_2) \ldots \circ F(\lambda_n),$$

where "$\circ$" denotes the composition of fuzzy relations. This means that if $S$ is a fuzzy relation on $X \times Y$ and $T$ is a fuzzy relation on $Y \times Z$, then $S \circ T$ is a fuzzy relation on $X \times Z$ and is defined by

$$S \circ T = R(x,z) = \bigvee_{y \in Y} \{(R(x,y) \wedge R(y,z)\},$$

where $\vee = \max$ and $\wedge = \min$. Hence $F^*(s, \lambda_1 \ldots \lambda_n, s') =$
$= \bigvee_{s'' \in S} (F^*(s, \lambda_1 \ldots \lambda_{n-1}, s'') \wedge F(s'', \lambda_n, s'),$
for any $(s, \lambda_1 \ldots \lambda_n, s') \in S \times \Lambda^* \times S$.

Figure 2 illustrates the method outlined in Step 3 of Algorithm 1 (cf. [7], Chapter 7). It shows that genetic algorithms are general enough to incorporate the evolution strategies of [9], and enables genetic algorithms to implement the strong causality philosophy of evolution strategies, where small changes of the cause must create small changes of the effect.

$$\begin{array}{|cc|cc|c|cc|} \hline a_1 & b_1 & a_2 & b_2 & \ldots & a_n & b_n \\ \hline \end{array}$$
$$\Downarrow$$
$$\begin{array}{|cc|cc|c|cc|} \hline b_1 & a_1 & b_2 & a_2 & \ldots & b_n & a_n \\ \hline \end{array}$$

Figure 2: Crossover operator with chromosomes as blocks for codewords.

A complete theoretical description of all minimal FSA of this sort is given by the following.

**Theorem 1** *The FSA* $\mathrm{Atm}_\ell(D, T, f)$ *is minimal if and only if* $\mu = \iota$, *that is* $C_0 = \{0\}$ *and* $x \neq y$ *implies* $\mathrm{In}(x) \neq \mathrm{In}(y)$, *for all* $(x, y) \in (T \times T) \cup (\overline{T} \times \overline{T})$.

*Proof* of Theorem 1. Let $D'$ be the subgraph induced in $D$ by the set $V' = V \cap f(X)$ of vertices. Consider the automaton $\mathrm{Atm}_\ell(D', T)$, defined by the graph $D'$ and the same function $f$. It is easily seen that $\mathrm{Atm}_\ell(D', T)$ recognizes the same language as the original automaton $\mathrm{Atm}_\ell(D, T)$. Further, we may assume that $V \subseteq f(X)$.

Every equivalence relation $\varrho$ on the set of states compatible with the next-state function defines the quotient automaton in a standard fashion. Since the initial state 1 is adjoined to every graph externally, in order to make sure that the quotient automaton is of the same format, we have to restrict our attention to equivalence relations such that the class containing 1 is a singleton. In this case we say that $\varrho$ is an *equivalence relation of the automaton* $\mathrm{Atm}_\ell(D, T)$.

The quotient automaton recognizes the same language if and only if the relation *saturates* the set $T$ of terminal states, i.e., $T$ is the union of some classes of the relation. Equivalence relations of this sort will be called *congruences* on the automaton $\mathrm{Atm}_\ell(D, T)$. This ensures that the quotient automaton is also defined by a graph. More formally, an equivalence relation $\rho$ on the set of states is called a *congruence* of $\mathrm{Atm}_\ell(D,T)$ if and only if it satisfies the following three conditions:

(C1) $(a,b) \in \rho$ implies $(a \cdot x, b \cdot x) \in \rho$, for all $a, b \in \mathrm{Atm}(D), x \in X$;

(C2) if $(a,b) \in \rho$ and $a \in T$, then $b \in T$;

(C3) the class containing 1 is a singleton.

Congruences of automata have been considered in the literature since they are important for studying efficiency of information processing algorithms, structure of FSA, optimization methods, and implementations of automata.

Denote by $\mathrm{Con}(D,T)$ the set of all congruences on $\mathrm{Atm}_\ell(D,T)$. Given congruences $\varrho, \delta$ on $\mathrm{Atm}_\ell(D,T)$, the meet $\varrho \wedge \delta$ and join $\varrho \vee \delta$ stand for their intersection and the transitive closure of their union, respectively. It is well known and easy to verify that the set of all congruences on any automaton forms a lattice with respect to $\wedge$ and $\vee$. Therefore $\mathrm{Con}(D,T)$ is a sublattice of the lattice of all equivalence relations on $V \cup \{0,1\}$ with 1 in a separate class.

The largest congruence on $\mathrm{Atm}_\ell(D,T)$ is the *Nerode equivalence* $\eta_T$ determined by

$$\begin{aligned} \eta_T \;=\; & \{(a,b) \in V \times V \mid a \cdot u \in T \text{ iff } b \cdot u \in T \\ & \text{for all } u \in X^*\} \cup \{(1,1)\} \end{aligned} \qquad (1)$$

(see, e.g., [8]). Denote the equality relation on $\mathrm{Atm}_\ell(D,T)$ by $\iota$. A congruence is said to be *proper* if it is distinct from $\iota$ and $\eta_T$. For any subset $S$ of $V$, denote by $\overline{S}$ the set $V \setminus S$. Note that $\overline{S}$ never contains 1. Clearly, an equivalence relation on $\mathrm{Atm}(D)$, with 1 in a separate class, saturates a subset $S \subseteq \mathrm{Atm}(D)$ if and only if it saturates $\overline{S}$. In order to consider the cases where $0 \in T$ and $0 \notin T$ simultaneously, we define the following set:

$$T_0 = \begin{cases} T \setminus \{1\} & \text{if } 0 \in T, \\ \overline{T} & \text{otherwise.} \end{cases}$$

The *in-neighbourhood* and *out-neighbourhood* of a vertex $v$ of $D = (V,E)$ are the sets $\mathrm{In}(v) = \{w \in V \mid (w,v) \in E\}$ and $\mathrm{Out}(v) = \{w \in V \mid (v,w) \in E\}$. We say that a subset $S \subseteq \mathrm{Atm}(D)$ is *in-closed* if $\mathrm{In}(S) \subseteq S$, where $\mathrm{In}(S) = \bigcup_{s \in S} \mathrm{In}(s)$. Putting $\mathrm{In}(0) = \emptyset$ we see that $\{0\}$ is in-closed.

Let $C_0$ be the set of all elements $c \in T_0$ such that there does not exist any vertex $v \in \overline{T}_0$ with a directed path from $v$ to $c$. Obviously, $C_0$ is the largest in-closed subset of $T_0$, and it always contains 0.

For any subset $S$ of $T_0$, consider relations

$$\begin{aligned} \mu_{S,S} &= (S \cup \{0\}) \times (S \cup \{0\}), \\ \mu_S^{T_0} &= \{(a,b) \mid \mathrm{In}(a) \cap \overline{S} = \mathrm{In}(b) \cap \overline{S} \\ &\qquad \text{and } a, b \in T_0 \setminus (S \cup \{0\})\}, \\ \mu_S^{\overline{T}_0} &= \{(a,b) \mid \mathrm{In}(a) \cap \overline{S} = \mathrm{In}(b) \cap \overline{S} \\ &\qquad \text{and } a, b \in \overline{T}_0\}. \end{aligned}$$

Define the relation

$$\mu_S = \{(1,1)\} \cup \mu_{S,S} \cup \mu_S^{T_0} \cup \mu_S^{\overline{T}_0}. \qquad (2)$$

Clearly, $\mu_S$ is an equivalence relation on $\mathrm{Atm}_\ell(D,T)$, and $\mu_S = \mu_{S \cup \{0\}} = \mu_{S \setminus \{0\}}$.

Denote by $0/\varrho$ the class of $\rho$ containing 0. It has been shown in [5] that an equivalence relation $\varrho$ on $\mathrm{Atm}_\ell(D,T)$ is a congruence if and only if $0/\varrho$ is an in-closed subset of $C_0$ and $\varrho \subseteq \mu_S$. Since the set $C_0$ is in-closed, it follows that the Nerode congruence $\sigma_T$ coincides with $\mu_{C_0}$ on $\mathrm{Atm}_\ell(D,T)$. An FSA is minimal if and only if its Nerode equivalence is the identity relation (see [2] or [8]). Thus, the FSA $\mathrm{Atm}_\ell(D,T)$ is minimal if and only if $\mu_{C_0} = \iota$, that is $\{(1,1)\} \cup \mu_{C_0,C_0} \cup \mu_{C_0}^{T_0} \cup \mu_{C_0}^{\overline{T}_0} = \iota$.

Clearly, $\mu_{C_0,C_0} = \iota$ if and only if $C_0 = \{0\}$. Further, assume that $C_0 = \{0\}$. Then the inclusion $\mu_{C_0}^{T_0} \cup \mu_{C_0}^{\overline{T}_0} \subseteq \iota$ turns into $\mu_{\{0\}}^{T_0} \cup \mu_{\{0\}}^{\overline{T}_0} \subseteq \iota$, and becomes equivalent to the following implication: for any $(x,y) \in T^2 \cup \overline{T}^2$, if $x \neq y$, then $\mathrm{In}(x) \neq \mathrm{In}(y)$.

Thus, it follows that the $\mathrm{Atm}_\ell(D,T)$ is minimal if and only if $C_0 = \{0\}$ and $x \neq y \Rightarrow \mathrm{In}(x) \neq \mathrm{In}(y)$, for all $(x,y) \in T \times T \cup \overline{T} \times \overline{T}$.

The definitions of a fuzzy automaton $\mathrm{fAtm}(D)$ and fuzzy congruence $\mu_S$ are similar to those of their crisp counterparts, and we omit them. In conclusion, let us formulate a description of fuzzy congruences on $\mathrm{fAtm}(D)$.

**Proposition 2** *Let $\rho$ be a fuzzy equivalence relation on* $\mathrm{Atm}(D,T)$*. Denote by $S$ the class of $\rho$ containing* $0$*. Then $\rho$ is a fuzzy congruence on* $\mathrm{Atm}(D,T)$ *if and only if $S$ is an in-closed fuzzy subset of $C_0$ and $\rho \precsim \mu_S$. In particular, for every in-closed fuzzy subset $S \precsim C_0$, the relation $\mu_S$ is a fuzzy congruence on* $\mathrm{Atm}(D,T)$*.*

*Proof.* The 'if' part: Suppose that $\rho \precsim \mu_S$ and $S$ is an in-closed fuzzy subset of $C_0$. Since $\mu_S$ satisfies conditions (C2) and (C3), it follows that the same can be said of $\rho$. In order to verify (C1) for $\rho$, consider any pair $(a,b) \precsim \rho$ and $f(x) = c$ where $c \not\precsim \{0,1\}$, i.e. $c \precsim V$.

First, if $c \precsim \mathrm{In}(a) \cap \mathrm{In}(b)$, then $(a \cdot x, b \cdot x) = (ca, cb) = (c, c) \precsim \rho$.

Second, if $c \not\precsim \mathrm{In}(a) \cup \mathrm{In}(b)$, then $(a \cdot x, b \cdot x) = (0,0) \precsim \rho$, too.

Third, suppose that $c \precsim \mathrm{In}(a) \setminus \mathrm{In}(b)$. We claim that $c \precsim S$. Indeed, if $a \precsim S$, then $c \precsim \mathrm{In}(a) \precsim \mathrm{In}(S) \precsim S$, because $S$ is in-closed. If, however, $a \not\precsim S$, then $\rho \precsim \mu_S$ implies that $\mathrm{In}(a) \cap \overline{S} = \mathrm{In}(b) \cap \overline{S}$, and $c \precsim S$ again. It follows that $(a \cdot c, b \cdot c) = (ca, cb) = (c, 0) \precsim S \times S \precsim \rho$.

The case where $c \precsim \mathrm{In}(b) \setminus \mathrm{In}(a)$ is similar, and so we have proved that (C1) holds. Thus $\rho$ is a fuzzy congruence on $\mathrm{Atm}_\ell(D,T)$.

The 'only if' part: Suppose that $\rho$ is a fuzzy congruence on the automaton $\mathrm{Atm}_\ell(D,T)$. Clearly, $S \precsim T_0$, because $\rho$ saturates $T$. To prove that $S$ is in-closed, take any vertex $a \precsim S$. Condition (C1) implies that $(b,0) = (ba, b0) \precsim \rho$, for every $b \precsim \mathrm{In}(a)$. Therefore $\mathrm{In}(S) \precsim S$.

In order to show that $\rho \precsim \mu_S$, pick any pair $(a,b) \precsim \rho$. If $a, b \precsim S$, then $(a,b) \precsim \mu_S$, because $S \cup \{0\}$ is an equivalence class of $\mu_S$.

Furthermore, assume that $a, b \not\precsim S$. Condition (C2) shows that $\rho$ saturates $T$, and so $a, b \precsim T_0 \setminus S$ or $a, b \precsim \overline{T}_0$. If there exists $c \precsim \mathrm{In}(a) \setminus \mathrm{In}(b)$, then (C1) implies $(c, 0) = (ca, cb) \precsim \rho$, and we get $c \precsim S$. Hence $\mathrm{In}(a) \setminus \mathrm{In}(b) \precsim S$. Similarly, $\mathrm{In}(b) \setminus \mathrm{In}(a) \precsim S$, and so $\mathrm{In}(a) \cap \overline{S} = \mathrm{In}(b) \cap \overline{S}$. By the definition of $\mu_S$ we see that $(a,b) \precsim \mu_S$. Thus $\rho \precsim \mu_S$. □

In conclusion let us note that the results above illustrate the possibility of using genetic algorithms and fuzzy automata to develop general methods of simplifying finite state automata, a classical concept of computer science crucial for numerous practical algorithms.

## References


[1] L. Davis, "Handbook of Genetic Algorithms", Van Nostrand Reinhold, New York, 1991.

[2] A.V. Kelarev, "Graph Algebras and Automata", Marcel Dekker, New York, 2003.

[3] A.V. Kelarev, S.J. Quinn, R. Smolikova, *On fuzzy regular languages*, AISAT'2000, Artificial Intelligence in Science and Technology, Hobart, Engineering, 291–295.

[4] A.V. Kelarev, O.V. Sokratova, *Languages recognized by a class of finite automata*, Acta Cybernetica 15 (2001), 45-52.

[5] A.V. Kelarev, O.V. Sokratova, *On congruences of automata defined by directed graphs*, Theoretical Computer Science 301 (2003), 31–43.

[6] J.N. Mordeson, D.S. Malik, "Fuzzy Automata and Languages", CRC Press, Chapman & Hall, 2002.

[7] M. Negnevitsky, "Artificial Intelligence: a Guide to Intelligent Systems", Harlow, Addison Wesley, 2001.

[8] G. Rozenberg, A. Salomaa (Eds.), "Handbook of Formal Languages", Vol. 1, 2, 3, Springer, New York, 1997.

[9] H.-P. Schwefel, "Evolution and Optimum Seeking", New York, Wiley & Sons, 1995.

[10] L.A. Zadeh, *Fuzzy languages and their relation to human and machine intelligence,* "Proc. Int. Conf. on Man and Computer", Bordeaux, France (1972), 130–165.